Claims

Sub A1

What is claimed is:

1) A method of scheduling CPU resources comprising the steps of:

a) using a counter to determine when to allocate the CPU resources;

b) instructing an interrupt controller to allocate the CPU resources; and

c) allocating the CPU resources.

2) The method of claim 1 wherein only a portion of the CPU resources are allocated.

3) The method of claim 1 wherein all of the CPU resources are allocated.

4) The method of claim 2 wherein the CPU resources are allocated to at least one thread, and the CPU resources are allocated by determining a duration of time and a periodicity for execution of said at least one thread.

5) The method of claim 3 wherein the CPU resources are allocated to at least one thread, and the CPU resources are allocated by determining a duration of time and a periodicity for execution of said at least one thread.

6) The method of claim 1 wherein the counter is a performance counter.

7) The method of claim 6 wherein the performance counter counts machine cycles in order to determine when to allocate the CPU resources.

a1 cont

8) The method of claim 6 wherein the performance counter counts executed computer instructions.

9) The method of claim 1 wherein the counter issues a first interrupt to the interrupt controller in order to instruct the interrupt controller to allocate the CPU resources.

10) The method of claim 9 wherein the interrupt controller issues a second interrupt to a CPU in order to instruct the CPU to allocate the CPU resources.

Sub a2

11) The method of claim 9 wherein the first interrupt is non-maskable.

12) The method of claim 10 wherein the first interrupt and second interrupt are non-maskable.

Sub a3

13) A method of scheduling resources on at least one microprocessor that includes a CPU and a device, the method comprising the steps of:

a) using the device to determine when to allocate the resources;

b) causing the device to issue a non-maskable interrupt to the CPU when it is time to allocate the resources; and

c) causing the CPU to allocate the resources in response to the non-maskable interrupt.

14) The method of claim 13 wherein the device is a performance counter.

15) The method of claim 13 wherein the device is a timer.

16) A method of scheduling resources on at least one microprocessor that includes at least one performance counter, at least one programmable interrupt controller and at least one CPU, said method comprising the steps of:

a) allowing the CPU to execute a first thread;

b) using the performance counter to determine when to allocate the resources to a second thread;

c) issuing a first non-maskable interrupt from the performance counter to the programmable interrupt controller when it is time to allocate the resources to a second thread;

d) instructing the programmable interrupt controller to issue a second non-maskable interrupt to the CPU that instructs the CPU to switch execution from the first thread to the second thread;

e) instructing the CPU to stop execution of the first thread;

f) causing the CPU to store first current state information regarding execution of the first thread;

g) causing the CPU to restore second current state information regarding execution of the second thread; and

h) allocating resources to the second thread.

17) The method of claim 16 wherein the programmable interrupt controller is an APIC.

18) The method of claim 17 wherein the microprocessor is selected from the group consisting of: a Pentium 4GB, a Pentium Pro 64GB, a Pentium MMX 4GB MMX, a Pentium II 4GB MMX, a Pentium III 4GB MMX KNI, a Celeron 4GB MMX, a Xeon PII 64GB MMX and a Xeon PIII 64GB MMX KNI.

19) A computer-readable medium having computer-executable instructions stored for performing steps comprising:

a) using a scheduler to control execution of at least one thread;

b) using at least one counter to notify the scheduler when to switch execution of said at least one thread.

20) The computer-readable medium of claim 19 further comprising instructions for issuing an interrupt from the counter when the counter reaches a predetermined number, said predetermined number defining a maximum duration for execution of said at least one thread, said interrupt notifying the scheduler to switch execution to another thread.

21) The computer-readable medium of claim 20 wherein said at least one counter is a performance counter and counts CPU cycles.

22) The computer-readable medium of claim 20 wherein said at least one counter is a part of a CPU and counts executed instructions.

23) The computer-readable medium of claim 20 further comprising instructions for executing said at least one thread at a highest IRQ level.

24) The computer-readable medium of claim 20 further comprising instructions for executing said at least one thread in a transparent manner so that at least one operating-system process is unaware of the execution of said at least one thread.

25) The computer-readable medium of claim 24 further comprising instructions for executing all of said operating-system processes and all of said at least one threads as a single real-time thread.

26) The computer-readable medium of claim 20 wherein the duration for execution of said at least one thread is not equal to the duration for execution of said another thread.

27) The computer-readable medium of claim 19 further comprising instructions for allocating at least a portion of a CPU's resources to an operating-system process and using the remaining CPU resources for execution of said at least one thread.

28) The computer-readable medium of claim 27 further comprising instructions for releasing the CPU resources back to the operating-system process when said at least one thread finishes execution.

29) The computer-readable medium of claim 27 further comprising instructions for releasing the CPU resources to another thread when said at least one thread finishes execution.

30) The computer-readable medium of claim 19 further comprising instructions for allocating a predetermined number of CPU cycles for execution of an operating-system process and using the remaining CPU cycles for execution of said at least one thread.

31) The computer-readable medium of claim 20 further comprising instructions for switching the interrupt from non-maskable to maskable.

32) The computer-readable medium of claim 20 further comprising instructions for switching the interrupt from maskable to non-maskable.

33) The computer-readable medium of claim 20 wherein said at least one thread and said another thread are of the same priority.